

AUSTRALIAN OS9 NEWSLETTER

Volume 8

August 1994

Number 6

EDITOR:	Gordon Bentzen	(07) 344-3881
SUB-EDITOR:	Bob Devries	(07) 278-7209
TREASURER:	Jean-Pierre Jacquet	(07) 372-4675
	Fax messages	(07) 372-8325
LIBRARIAN:	Bob Devries	
CONSULTANT:	Don Berrie	(079) 75-3537
SUPPORT:	Brisbane OS9 Users Group	

CONTENTS

OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9
OS9								OS9
OS9								OS9
OS9	Editorial	Page 2						OS9
OS9	Programming in Basic(09).....	Page 3						OS9
OS9	Tuning a Burke & Burke H.D.....	Page 5						OS9
OS9	OS-9 P.D. Library	Page 8						OS9
OS9	INFO ! by Rod Holden	Page 8						OS9
OS9	UltiMuse III users	Page 10						OS9
OS9	more INFO!	Page 11						OS9
OS9	OS-9 Usergroup Member list	Page 12						OS9
OS9								OS9
OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9

EDITORIAL MATERIAL

Gordon Bentzen
8 Odin Street
SUNNYBANK QLD 4109

LIBRARY REQUESTS

Bob Devries
21 Virgo Street
INALA QLD 4077

AUSTRALIAN OS9 NEWSLETTER
Newsletter of the National OS9 User Group
Volume 8 Number 6

EDITOR : Gordon Bentzen
SUBEDITOR : Bob Devries

TREASURER : Jean-Pierre Jacquet
LIBRARIAN : Rod Holden

SUPPORT : Brisbane OS9 Level 2 Users Group.

Greetings to all,

This is it, the very last, last newsletter from the Australian OS-9 Usergroup. I cannot help but reflect on the last six years and wonder how we managed, with a few exceptions, to produce a newsletter each month.

It all started for us back in mid-1988 when three members of the local Brisbane, Queensland, OS-9 Usergroup decided to take on the task of reviving the "National OS9 Usergroup" which wound up some six months earlier.

So Don Berrie, Bob Devries and myself began the planning and production of the first "new" trial newsletter. This was mailed out in July 1988 to known OS-9 users. The response was encouraging so we pressed on (pun intended) and took subscriptions beginning September 1988.

Now I must admit that if it wasn't for the OS-9 knowledge and enthusiasm of Don Berrie and Bob Devries this whole project would have been very short lived. None of us, however, had a vision of this continuing for anything longer than a year or two.

We have been very proud to support OS-9, mostly the Level 2 version for the CoCo3, and are grateful for the support, contributions and suggestions from many of our members over the past six years.

A good deal of debate has taken place over those years about the future and "place" for OS-9 and this has been evident most recently on the Internet listserver as well as a number of other forums. I can also recall our concern at the news that Tandy was to drop the CoCo. Was this to be the end of the OS-9 Usergroup? - Well, NO.

I do not intend to take up that debate in this editorial but simply state that I, for one, have learnt a lot through my association with OS-9 and OS-9 users. It certainly holds true, in this case, that "the more one learns, the more one realises how much more there is to learn".

In this edition we have a number of very interesting articles, including one by Peter Edwards on the Burke

& Burke hard drive system for the CoCo. Bob Devries makes comment about the future of our P.D. library, and by the way, the P.D. stuff from Hawaii has now arrived.

There are some thirty odd 1.44MB disks from the OCN, OS-9 Community Network. Bob and a couple of helpers are at present sorting through these disks to see what we have got. Some of the files we already have, so some cross reference will be necessary.

Also included with this edition is our membership list, complete with mailing addresses, so that those interested may have a means of maintaining contact with other OS-9 users.

Bob Devries tells me that he has Email messages from the U.S. OS-9 Usergroup requesting a copy of our mailing list. Here it is! We do plan to stay in touch and look forward to hearing about any new developments.

It is with mixed feelings that we prepare this final newsletter now that the decision has been made to stop. Once a month comes around very fast as I am sure you are well aware. A monthly newsletter is put together, printed, addressed and mailed, then, somebody says "it's time to start thinking about our newsletter again". But didn't we just do one?

So, it will be a relief not to have to try for that monthly deadline all the time, but I am sure that we will miss this also.

We are not going to pack up OS-9 and store it in a cupboard somewhere, we will continue to use it, so all is not lost.

Finally, we are grateful for the support of many members who have continued to subscribe to the National OS9 Usergroup even though they have moved on to other operating systems and of course thanks to those members who have been active in this usergroup.

We wish you all the very best in your future endeavours and trust that you continue to enjoy your computing.

Farewell, Gordon Bentzen.

Programming in Basic(09)
by Bob Devries

I was recently asked to convert an RSDOS basic programme to work under OS-9's Basic09. It was only fairly short, so presented no problems in itself.

Nowadays, when I do any development work, I do so with my SECAD OSK computer, and Basic programming is no exception. I have Microware Basic for my SECAD AS-68K, and I have found that programmes written for Basic09 and MW Basic are interchangeable at source level.

However, as always, there are SOME exceptions. MW's Basic for OS-9/68000 was written in C (as opposed to Basic09 which was written in assembler), and so it uses the C convention of terminating strings with NULL byte (\$00). Basic09 uses a \$FF byte. Neither of them appear to use the length of string system that RSDOS basic uses internally.

The MW Basic can cause problems with this method of string termination, especially if you want to send control code strings to the printer.

I tried to be a NICE programmer, and set up the control codes in a string variable, like this:

```
DIM control:STRING
```

```
control = CHR  
$( $1b)+CHR$( $  
5b)+CHR$( $40)+CHR$( $00)+CHR$( $00)+CHR$( $02)....
```

The problem here is, MW Basic gets to the first \$00, and figures that it's found the end-of-string, and stops sending anymore characters, regardless of whether I use PRINT or PUT.

If I use Basic09, a similar thing happens if I use a \$FF code in a printer control string.

Here's a little Basic PROCEDURE which I used to get around the problem. It is equally at home in Basic09 as in MW Basic.

To use this procedure, first open a path to your printer:

```
DIM printer:integer  
OPEN #printer,"/p":WRITE
```

Then, to use, for example, double strike mode, run

the procedure like this:

```
RUN ptrctl(printer,"doublestrike","on")
```

Don't get the spelling wrong, or it won't work. Also, the 'doublestrike' and 'on' MUST be in lowercase, else it won't work. If, for example, you type 'ON' (in capitals) the procedure won't recognise it, and do the 'off' code. To make this code case-insensitive, would take a bit more code.

Bob Devries

```
PROCEDURE ptrctl  
PARAM path:INTEGER  
PARAM action:STRING  
PARAM onoff:STRING  
DIM number:INTEGER  
DIM count:INTEGER  
DIM char:BYTE  
DIM found:BOOLEAN
```

```
(* PTRCTL - send printer control character *)  
(* strings to printer *)  
(* By Bob Devries. *)  
(* InterNet: bob@splat.paxnet.com.au *)  
(* PUBLIC DOMAIN - 22nd July, 1994 *)
```

```
found = FALSE  
IF action = "expand" THEN  
    found = TRUE  
    IF onoff = "on" THEN  
        RESTORE 100  
    ELSE  
        RESTORE 110  
    ENDIF  
ENDIF  
IF action = "underline" THEN  
    found = TRUE  
    IF onoff = "on" THEN  
        RESTORE 120  
    ELSE  
        RESTORE 130  
    ENDIF  
ENDIF  
IF action = "condensed" THEN  
    found = TRUE  
    IF onoff = "on" THEN  
        RESTORE 140  
    ELSE  
        RESTORE 150  
    ENDIF
```

AUSTRALIAN OS9 NEWSLETTER

```
ENDIF
IF action = "compressed" THEN
    found = TRUE
    IF onoff = "on" THEN
        RESTORE 160
    ELSE
        RESTORE 170
    ENDIF
ENDIF
IF action = "doublestrike" THEN
    found = TRUE
    IF onoff = "on" THEN
        RESTORE 180
    ELSE
        RESTORE 190
    ENDIF
ENDIF
IF action = "emphasized" THEN
    found = TRUE
    IF onoff = "on" THEN
        RESTORE 200
    ELSE
        RESTORE 210
    ENDIF
ENDIF
IF action = "doubleheight" THEN
    found = TRUE
    IF onoff = "on" THEN
        RESTORE 220
    ELSE
        RESTORE 230
    ENDIF
ENDIF
ENDIF
READ number
FOR count = 1 TO number
    READ char
    PUT #path,char
NEXT count
END

(* The first data statement is the *)
(* number of code bytes *)
(* that follow for the selected code *)

(* These codes are for a Tandy DMP202 *)
(* (IBM ProPrinterXL compatible) *)

(* "expand" "on" code *)
100 DATA 3,$1b,$57,$1
(* "expand" "off" code *)
110 DATA 3,$1b,$57,$0
(* "underline" "on" code *)
120 DATA 3,$1b,$2d,$1
(* "underline" "off" code *)
130 DATA 3,$1b,$2d,$0
(* "condensed" "on" code *)
140 DATA 1,$f
(* "condensed" "off" code *)
150 DATA 1,$12
(* "compressed" "on" code *)
160 DATA 2,$1b,$3a
(* "compressed" "off" code *)
170 DATA 1,$12
(* "doublestrike" "on" code *)
180 DATA 2,$1b,$47
(* "doublestrike" "off" code *)
190 DATA 2,$1b,$48
(* "emphasize" "on" code *)
200 DATA 2,$1b,$45
(* "emphasize" "off" code *)
210 DATA 2,$1b,$46
(* "doubleheight" "on" code *)
220 DATA 9,$1b,$5b,$40,$04,$0,$0,$0,$22,$2
(* "doubleheight" "off" code *)
230 DATA 9,$1b,$5b,$40,$04,$0,$0,$0,$11,$1
```

oooooooooooooooooooooooooooooooo

AUSTRALIAN OS9 NEWSLETTER

Tuning a Burke & Burke Hard Disk Drive by Peter Edwards

This case history of tuning up a B&B-connected hard drive is offered in the hope that it might help someone else who is about to try it. Experts are encouraged to point out the errors, and (in particular) to offer explanations of the interleaving measurements described at the end.

After getting the B&B running, thanks to Andrew Donaldson, I decided to check that it was wasting as little space as possible, and running as fast as possible. The only specifications I had were the results of running performance programs on a PClone, before the drive was removed for use with the CoCo. This claimed that the drive, a CDC StorageMaster 518, had 3 heads, 495 cylinders and 17, 512 byte sectors per track (ie. 34, 256 byte ones).

First of all, I rebuilt the descriptor assuming one extra cylinder. It worked! So I got greedy and retried with another one (ie. 2 more than was used on the PClone). FORMAT finished, WITHOUT ANY MESSAGE, immediately after accepting the disk name. (BTW, you are aware, aren't you, that many 40 track floppy drives can go to 41 or 42 tracks? All you have to do is patch the descriptor, and remember not to use such disks for sending to friends.)

Buoyed up by this success, I tried 36 sectors per track, two more than MessDOS uses, and four more than B&B suggest. This time, FORMAT failed with error 241 - Bad system sector, FORMAT aborted. Fair enough -

B&B Step-code ('xx' above)	0	1	2	3	4	5	6	7
Step-rates (microsec)	3000	45	60	18	200	70	30	18
COPY time (seconds)	32	33	32	32	32	32	32	33
CMP time (seconds)	54	55	54	46	47	46	54	54

Clearly, using a COPY command to measure step-rates is no good; COPY's buffers must so big that it rarely needs to switch between files. On the other hand, CMP must use small buffers, perhaps just a sector or two for each file. It therefore swaps furiously between the two files. This produces timing differences which stand out clearly. For MY drive, not necessarily YOURS, I chose a step code of 5, which happens to be B&B's recommendation for those who don't know what to do. Note that the step-CODES are not in step-RATE order.

The last tests were for the optimal interleave (EXPERTS, PLEASE TUNE IN HERE). The interleave is only honoured when the drive is formatted. In normal

it must have plonked the last two sectors on top of the first one or two of the same track. (Note that there is no point in trying an odd number of OS-9 256 byte sectors, as B&B pack them in pairs into the 512 byte sectors as used by MessDOS.)

Then I tried with 4 heads, instead of 3, a real long shot. Drive manufacturers don't leave spare heads and disk surfaces inside their products! -- FORMAT hung.

At this point, I figured I had squeezed as much capacity as possible out of it. Not too bad; the extra two sectors was obvious, but the extra cylinder was a nice surprise. Guess the moral is "Don't believe all you're told".

Now for speed. First of all, the step-rate. My benchmark was the following line:

```
DMODE /hd stp=xx; TIME COPY x.ar y.ar; TIME CMP  
x.ar y.ar; DEL y.ar
```

All commands had been LOADED into memory. File x.ar was created by running the contents of my CMD5 directory through AR; any decent sized file will do. DMODE is on the List; TIME is on Delphi. Someone should get permission to put it on the List.

I got the following average results (don't fuss about the numbers; just look at the trends):

use, it is ignored, as the sector headers contain the sector number, and that is what is important. THIS TEST THEREFORE DESTROYS THE DATA ON YOUR DISK. It is also bloody slow, due to the need to reformat the disk for each test.

The benchmark command was:

```
DMODE /hd ilv=xx; FORMAT /hd; FREE /hd; TIME MEGAREAD  
</hd@
```

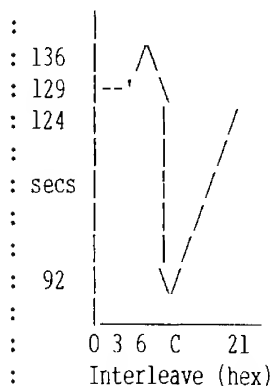
(The "FREE" command is included to force a seek to track zero.) This was done with a temporary device descriptor which cut down the number of cylinders to describe a disk of about 2MB (69 cylinders for mine).

This is to minimize the time spent reformatting. Once again, all commands were memory resident. Like TIME, MEGAREAD is from Delphi. I won't give the figures for this, as I did it for every second value between 0 and \$1E (DMODE uses hexadecimal!). These were graphed, and the dip in the graph was confirmed by checking the odd-numbered interleaves, and by remeasuring where it looked funny. All in all, 33 times :- (The graph looked like that shown in Fig 2 below.

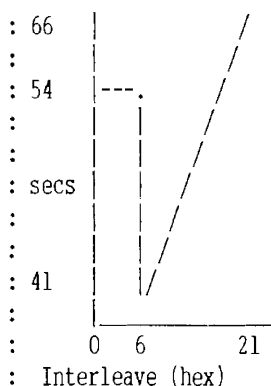
The key features are:

Horizontal from interleaves from 0 to 3 at 54 secs
 Near-vertical from 4 to the minimum of 6, where it got down to 40 secs
 Sloping line from 6 to \$1E, where it was 65 secs

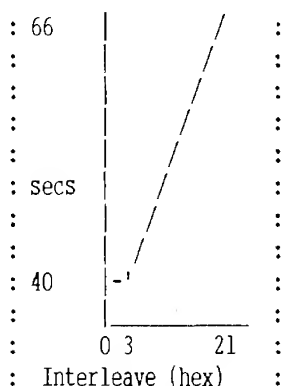
I was unhappy with a few aspects of this, so I measured again with different commands. Because a large proportion of sequential multisector disk I/O, which is when interleaves matter most, is in the loading of binary modules into memory, I felt a benchmark involving loads should be tried too. So I wrote a Mickey Mouse program ("MEGALOAD") which repeatedly loads a large module from disk. The whole measuring process was then repeated.



FORMAT
 Fig 1



MEGAREAD
 Fig 2



MEGALOAD
 Fig 3

Now for "TIME MEGAREAD </hd@" (Fig 2). Note that there is no hump.

Finally, "TIME MEGALOAD" (Fig 3). Note, no plateau or cliff.

The best spot (the start of the straight line on the right), varies: 3 for MEGALOAD, 6 for MEGAREAD (right where FORMAT's hump is) and \$C for FORMAT. This is what we expect, considering the different amount of

work being done after each read. The optimal spot according to FORMAT has a 20% performance hit if it's MEGALOAD which is telling the truth.

```

l os9pi
c 05b7 cc 4f
c 05b8 00 5f
c 05b9 02 39
v
  
```

speeded it up, and made it a better test of disk read performance. (I run without CRCs all the time now, because of the performance improvement. If my disk goes bad, loading a dud module or two is the least of my worries!)

First, the results of (if I remember correctly)

```
ECHO yyy ! TIME FORMAT /hd "dummy name" -r
```

or whatever params you need to have the whole thing run without any interactive prompts. See Fig 1. Note the small but definite hump between the plateau and the cliff.

work being done after each read. The optimal spot according to FORMAT has a 20% performance hit if it's MEGALOAD which is telling the truth.

It would be interesting to alter MEGAREAD, so instead of reading in 1MB in 1024B chunks, it used 256 byte (ie. one logical disk sector) or 512 byte (one physical sector) instead. I have a feeling it would be closer to MEGALOAD then. I think the load system calls would read in single sector units, and probably

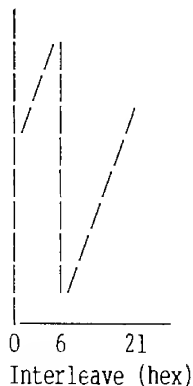
direct to the data's final location. C programs probably read into an RBF buffer first, then get copied to the user's program data area, and in the case of MEGAREAD, re-blocking would be needed as well. Unforch, I can't try this - my compiler is on my backup floppies, and I don't feel like reloading them, modding MEGAREAD, then scrapping the HD again!

I guess the hump in the FORMAT curve is because I am measuring the format time as well as the verify time; I probably have the sum of two quite different curves. A low-grade benchmark. A pity, because it's so easy.

Anyway, I chose an interleave of 7, just to get a bit of distance from the vertical section. (B&B recommend a value of 22 (\$16).)

However, I am unhappy about a few aspects of these measurements. At 6, the next sector arrives at the read-head just when MEGAREAD wants it. At 7, it has to wait for one sector time; at 8, it waits for two sector times. So far, so good - this explains the straight line as the interleave increases past a value of 6. Going the other way, at an interleave of 5, the next sector has just been missed, and an entire revolution has been wasted. Hence the vertical bit.

My problem is with the horizontal portion. I would have thought that going to an interleave of 4 means you waste one revolution, less one sector, and going to 3 would mean 1 rev less 2 sectors etc. This would produce a graph like:



This would mean that the graph wraps around.

Any ideas anyone? I have a few, but I'm unsure what to make of them:

One is the fact that pairs of OS-9 256 byte sectors are packed into single 512 byte ones before being written. I assume that reading consecutive 256 byte sectors results in a disk access on every second 'logical' read. Or is a 512 byte sector read each time, with half of the data being discarded?

What happens when you write randomly? Sequentially? Do you get 'lost' revolutions when writing a 256 byte sector due to the necessity to read its neighbour first?

Do PC disk controllers include some sort of buffer, which acts as a cache at low interleave values but not at high ones? (Off the wall, this one.)

At the time of writing, I have not had time to try asking the experts.

The MEGALOAD source and its build script (bld) have been tested with both asm and rma. In the source, the assembler-specific lines are marked in the right margin. Just (un)comment the ones you need. C.asm & c.link should be the same as rma & rlink - I haven't tested them.

As written, the file which is repeatedly loaded is /h0/compiler/c.pass1, which is just pass 1 of the C compiler - I'm sure you've got that! I chose it because

- it's one of the biggest modules I have (31kB I think)
- most people who would want to tune a HD will have it
- it's a single module file
- the module name matches the filename

In any case, anyone who doesn't like it can edit the source and re-assemble.

oo

AUSTRALIAN OS9 NEWSLETTER

The future of the PD Library

by Bob Devries

As you are no doubt aware, this is your last newsletter from us. With that, of course, all announcements of new Public Domain or commercial software for OS-9 will also cease. It is, however, my intention to continue to maintain the PD library, and even to expand it. Currently we have all of the US OS-9 library, our own 11 disk collection, and we are waiting for some 25 to 30 1.44MB disks to be returned from Hawaii, with the library of the OS-9 Community Network.

While I can't promise to maintain a complete commented list of the software, I can, and will supply to anyone who asks (and sends a disk, as before) a list of the names of the files. I realise that this is not as useful as the database, and the commented list from the RiBBS files.bbs file, but it is at least something.

For those who are interested, and who are willing to keep up a communication with me, I will send the

necessary information about updates. Perhaps a three monthly mailing to interested OS-9 users would be a start.

I will be downloading whatever new software is available for OS-9/6809, OS-9/68000, and OS9000, from the InterNet FTP sites, and including these in the PD library.

The PD library also contains about 4 MB of RSDOS software. These files have actually always been there, but I haven't made an issue of it before.

Interested people should contact me at the address below, or phone me.

Bob Devries
21 Virgo Street,
INALA. Qld. 4077

PH: (07) 2787209

oooooooooooo0000000000oooooooooooo

```
IIIIII NNNN  NN FFFFFF 000000 !!
  II  NN NN  NN FF  00 00 !!
  II  NN NN  NN FFFF  00 00 !!
  II  NN NN  NN FF  00 00 !!
  II  NN  NN NN FF  00 00
IIIIII NN  NNNN FF  000000 oo
```

By Rod Holden

Hi, and welcome to Info. This particluar piece of software is in a file called Qtip4l.ar which is available under the OCN OS9_UTI directory, please read on;

* * Q T I P * *

Written by Frans Lichtenberg

(c) 1988,1989,1990 CFL Software

A disk management utility for the Colorcomputer 3 under the OS9 Level II operating system.

INTRODUCTION

Qtip was written out of need for a easy to use disk editor that was capable of aiding in the repair of a disk too. Everything has been made to be operated with the arrow keys and the spacebar, except for the places you are asked to enter a string, decimal or a hex value.

What is required: QTIP, the main program.

STARTING QTIP

QTIP can be loaded with or without an argument. If omitted QTIP will prompt for a filename when the main screen is displayed. If a '?' is used a an argument a help message will appear. The command line for QTIP can look like this:

qtip /h00 --> Open the hole media.

AUSTRALIAN OS9 NEWSLETTER

qtip filename --> Open file in current directory.
qtip /d0/cmds/filename --> Open file on drive 0.
qtip --> Enter argument on main screen.
qtip -? --> Display usage.

The first command line will open the whole disk, raw mode. The second and 3rd command line is two different ways to open an file. The 4th command QTIP will ask for a filename on the main screen. The last will display a usage message. Later I will explain how you can change files and/or raw mode on the fly.

The first sector (raw or file mode) will always be displayed. In the top left corner you will find the filename displayed and in the right corner the sector number in hex and decimal. The main menu and the sub-menu is displayed in the most right column. The bottom field is used for displaying messages and input sector numbers and strings for search. In the middle of it all the sector is displayed, both in hex and ascii. In the lower right corner the last sector number (raw/file) is displayed.

COMMANDS

Next/Prev

By using the up and down arrows, as indicated in the command field, place the inverted field on NEXT/PREV. Hit the spacebar and the next or previous sector will be displayed. You can do it again and again until you found the sector you need. The same goes for PREV it just works the other way.

Copy

Place the menu-bar over COPY by using the up/down arrows and hit spacebar. This will send a copy of the displayed sector to a printer. There are no control characters involved so any printer can be used.

NewS(ector)

Place the menu-bar over NEWS and hit the spacebar, the DEC in the sub-menu will be inverted and you can use the up/down arrows to alter between DEC and HEX, now you can hit the spacebar and in the command field you are asked to enter a new sector number in hex or decimal depending of your selection in the sub-menu. There are two ways you can enter the new sector number, either using all five decimals

with leading zeros, the last digit will automatically start the command or just enter the new sector number and round it of with an ENTER. The screen will settle again just like when you started up the program but displaying the new sector.

CHaNGe

Place the menu-bar over CHG and hit the spacebar, the ASCII in the sub-menu will be inverted and you can use the up/down arrows to alter between ASCII and HEX, hit the spacebar and a cursor will appear in the HEX or ASCII display area in the upper left corner. Use the arrow keys to move the cursor. When replacing values in the HEX area, a single character in the left nibble will be moved to the right nibble when saved. To avoid this you must always enter the right nibble to get the proper result. To exit you press the ALT-Q on the keyboard. The sector is NOT saved. A message in a window will display a reminder to that effect.

SAVE

Place the menu-bar over SAVE and hit the spacebar. The sector displayed will be saved on disk at its proper place. A message in a window will tell you that the sector HAS been saved.

APND

Selecting 'Apnd' will prompt the use on the command line for a full pathname. Any device (disk drive) can be used. Apnd will then save the displayed sector, using the pathname. The path will be closed. A second sector can be appended if using the same pathname. A window will be displayed indicating that the save was successful.

FIND

Place the menu-bar over FIND and hit the spacebar. You will be prompted on the command line to enter a ascii string (max 16 characters) to search for. If the string is found the sector will be displayed. The user will be prompted to find the next occurrence of the string. The search can be stopped by pressing any key on the keyboard and the sector where it stopped will be displayed. If no match the last sector will be displayed.

FILE

Place the menu-bar over FILE and hit the spacebar. If the file has been corrected it will

AUSTRALIAN OS9 NEWSLETTER

have a new CRC calculated. You will be prompted on the command line to enter a new filename. If file found the beginning of that file it will be displayed.

QUIT

Place the menu-bar over QUIT and hit the spacebar. If you have made a SAVE the file will be verified and restored on disk, and QTIP will terminate.

(T) = TRANSLATE

By hitting the key 'T' on the keyboard, when displaying either the Logical Sector Number 0 (LSNO), a directory sector, the File Descriptor (FD) or the first sector in a executable module, a window will appear with a translated explanation of the contents of the sector.

Logical Sector Number 0

This window will display all information regarding the media, number of byte and sectors. The location of the start of ROOT directory, the start and size of the File Allocation Table (FAT), and much more. This window is only available if the 'Raw Mode' option has been used when Qtip was started.

Directory Sector

This window is more for easy reading of the directory sectors. It will display up to 8 filename entries and the corresponding File Descriptor sector address. Filenames that have been deleted and not overwritten will be displayed with the first character in the filename replaced with an '*'. The Directory Sectors will also only be displayed if Qtip has been started with the 'Raw Mode' option set.

File Descriptors

The File Descriptor window will display the size of the file and the creation date. The main thing with this display is the Segment Address Map. Five entries can be displayed despite the sector has room for 48. Again this window can only be activated if Qtip has been started with the 'Raw Mode' option.

File Header Sector.

This is the first sector in a file (module). All the relevant information will be displayed including the filename. This window can be used both in file-mode and raw-mode, if the file is executable.

(H) = HELP

Pressing 'H' will display a window with all the commands explained.

(A) = ASCII TABLE

Pressing 'A' will open a window where the ascii table will be displayed. Use the ENTER key to scroll thru the table.

(D) = DIRECTORY

Pressing 'D' will display your data directory in a window. Pressing any key will close it.

(S) = Sector Lock Out

Pressing 'S' will ask on the command line if you want to change a sector value. If 'n' is pressed you will return to the main menu. If a 'y' is pressed you will be asked if you want to 'Remove' or 'Add' a sector. You will then be asked to enter a sector number in hex, when enter is hit the appropriate bit in the sector map will be set or cleared.

Bye for now.

oooooooooooo0000000000oooooooooooo

UltiMusE III Users, Read this!

This is for anyone who currently owns a copy of UltiMusE III for the Colour Computer 3, and would like to get an upgrade to the latest version, 4.10.0.

I have been given permission by the author, Mike Knudsen, to distribute upgrades to users of UltiMusE III in Australia.

You supply the **ORIGINAL** disk, in a re-usable mailer, with return postage, and I will copy the new version of UltiMusE III onto it. Please **WRITE** to me, enclosing your **NAME**, and **ADDRESS**, and the original UltiMusE disk. I require you to send me your **ORIGINAL** disk as proof of purchase.

AUSTRALIAN OS9 NEWSLETTER

AMBROSI	G. A.	172 OGILVIE STREET	ESSENDON	VIC 3040	
BAILEY	Eric	61 WINCHESTER STREET	MOONEE PONDS	VIC 3039	
BAMFORD	Clyde R	23 RAEBYRN ROAD	ROLEYSTONE	WA 6111	
BARKER	Robert	P.O. BOX 223	GLENFIELD	NSW 2167	
BARTOLEC	Dubravko	13 MCKAY STREET	DUNDAS VALLEY	NSW 2117	
BENTZEN	Gordon	8 ODIN STREET	SUNNYBANK	QLD 4109	
BERRIE	Don	"EAST END" MS 1445	MT LARCOM	QLD 4695	
BLAZEJEWSKI	Stan	51 EVAN STREET	PARKDALE	VIC 3195	
BOARDMAN	Margaret	U2/26 MONASH RD	PORT LINCOLN	SA 5606	
BYE	Graham	9 AIRLIE BANK RD	MORWELL	VIC 3840	
COSSAR	M. E.	12 RAKEIORA GROVE	PETONE	6008	NEW ZEALAND
CUNNINGHAM	Eric	7 NUTHATCH STREET	INALA	QLD 4077	
DEVITT	DENNIS	21W 144 CANARY ROAD	LOMBARD		U.S.A. IL 60148
DEVRIES	Bob	21 VIRGO STREET	INALA	QLD 4077	
DONALDSON	Andrew	5 THE GLADES	DONCASTER	VIC 3108	
DONGES	Geoff	P.O. BOX 326	KIPPAX	ACT 2615	
EATON	David	20 GREGSON PLACE	CURTIN	ACT 2605	
EDWARDS	Peter	40 DAVISON STREET	MITCHAM	VIC 3132	
GALL	Brian D	PO Box 131	COORANBONG	NSW 2265	
GODFREY	Stephen	HEINEMANN ROAD	REDLAND BAY	QLD 4165	
HARBECKE	Peter	18 MACHENZIE STREET	MANLY WEST	QLD 4179	
HOLDEN	Rod	53 HAIG ROAD	LOGANLEA	QLD 4131	
HUTCHINSON	Simon	10 ASCOT COURT	NTH DANDENONG	VIC 3175	
IKIN	John C	42 SPRUCE DRIVE	ROWVILLE	VIC 3178	
JACQUET	J.P.	27 HAMPTON STREET.	DURACK	QLD 4077	
JOHNSON	FRASER	35 ROBSON AVE	GOROKAN	NSW 2263	
KINZEL	Burghard	LEIPZIGER RING 22A	5042 ERFTSTADT		GERMANY
LARAWAY	Terry	41 N.W. DONCEE DRIVE	BREMERTON		U.S.A. WA 98310
MACKAY	Rob	7 HARBURG DRIVE	BEENLEIGH FOREST	QLD 4207	
MacKENZIE	Greg	80 CRESWELL STREET	CRIB POINT	VIC 3919	
McGRATH	A. John	93 LEMON GUMS DRIVE	TAMWORTH	NSW 2340	
McLINTOCK	George	7 LOGAN STREET	NARRABUNDAH	ACT 2604	
McMASTER	Brad	P.O. BOX 1190	CROWS NEST	NSW 2065	
McNABB	John	PO BOX 109	BORONIA	VIC 3155	
MURPHY	Kevin	108 ADENEY AVE	KEW	VIC 3101	
PALMER	Brian	9/2 PARA STREET	BALGOWNIE	NSW 2519	
REMIN	Fred	100 WHITSUNDAY DRIVE	KIRWAN	QLD 4817	
SINGER	Maurice	217 PRESTON ROAD	WYNNUM WEST	QLD 4178	
SKEBE	Jeff	23 NORMA ROAD	PALM BEACH	NSW 2108	
STEMAN	John	P.O. BOX 680	WINDSOR	NSW 2756	
SWINSCOE	Robin	17 MELALEUCA STREET	SLADE POINT	QLD 4740	
TURNER	Peter	13 PARKWOOD GR.	MITCHELL PARK	SA 5043	
TUTELAERS	Peter	STRIJPERSTRAAT 50A	5595 GD LEENDE		NETHERLANDS
UNSWORTH	Rob	20 SALISBURY ROAD	IPSWICH	QLD 4305	
USERS GROUP, Inc	The OS-9	6158 W. 63rd Street	Suite 109	U.S.A.	CHICAGO ILL 60638
RAE	Desmond	P.O. BOX 2076	MOUNT ISA	QLD 4825	